Abstract—Databases have been an integral component of Data Fusion from the outset when the JDL model was introduced. As advances in High-Level fusion using Multi-Int data have been made, the original concept of databases as a static repository of Level 0/1 content has evolved to support heterogeneous data, and as a necessary enabler of High-Level fusion processes. Relatively recent database technologies now support specialized storage for complex content such as multi-media, geospatial, and semantic data types. Additionally, database functionality has been extended from what was once almost exclusively storage and retrieval, to include integrated forensic and predictive algorithms, as well as decision support frameworks such as the data cube. These data mining capabilities provide a rich tool-set from which to tailor a fusion application. However, due to their inherent trade-off space, they present a significant design and integration challenge when implementing an enterprise architecture, which has to provide a comprehensive and cohesive framework across the entire fusion workflow, and which has to meet the needs of various Communities-of-Interest. This paper expounds on the role of data architecture as a key discipline to help analyze and synthesize an enterprise fusion system-of-systems, and presents selected principles to maximize heterogeneous data exploitation.

Keywords—Data Mining; High-Level Fusion; Data Analytics.

I. INTRODUCTION

The classic Joint Directors of Laboratories (JDL) Fusion model comprises considerations for a database [1], which continues to be recognized as a critical component. Waltz describes data fusion and data mining, along with workflow processes such as data cleansing and Extract Load and Transform (ETL), and introduces the Data Warehouse as part of the overall fusion architecture [2]. The complexity inherent in supporting Multi-Int data, e.g., OSINT, HUMINT, SIGINT, IMINT, GEOINT throughout the fusion exploitation workflow, necessitates a flexible architectural paradigm that supports the various algorithm, modeling, and format requirements as data is transformed into information, information into knowledge, and knowledge into cognition [3, 4].

Salerno highlights the extent to which data management and data mining are coupled with the fusion processes, and discusses techniques and challenges for knowledge representation, data normalization, structuring, and cleansing [5-7]. The importance of data mining and predictive analytics has not been lost on the Fusion community; McCue highlights how these tools can help in the analysis of a data saturated environment [8, 9]. Notwithstanding, there remains a need to integrate these capabilities into an Enterprise High-Level (HL) Fusion Architecture. Consider Blasch’s addition of Mission and User views to the original JDL model, which more comprehensively describe a mission-driven impetus of top-down data management requirements for developing a Course-of-Action (COA) [10]. Accordingly, an enterprise architecture imposes top-down mission objectives and bottoms-up collection platform data management requirements across diverse Communities-of-Interest (COIs), which need to be reconciled. Fig. 1 depicts a mission centric view of Decision Fusion (DF) with the objective to generate a COA that enables Blue Forces (e.g., COIs) to mitigate, neutralize, or optimally prevent negative impacts of Red Forces (e.g., adversaries, crime, natural disasters). The characterization and abstraction layer depicts that Multi-Int data primitives undergo structural mutations throughout the forensic and predictive cycles. The objective of this paper is to present data architecture principles based on existing technologies as design components that help maximize data exploitation throughout the fusion process.

Figure 1. Data characterization and mutation throughout the fusion process
II. THE ROLE OF ENTERPRISE DATA ARCHITECTURE

An Enterprise Architecture (EA) aligns processes and implementation components with the Mission Objectives. This System-of-Systems comprising distributed collection platforms (producers) and fusion nodes (consumers) presents challenges that stem from the need to reconcile often technically incompatible approaches and the diverse business drivers inherent in integrating multiple COIs. DoDAF and the Zachman framework provide comprehensive coverage of Enterprise Architecture concepts and design [11-13]. Fig. 2 illustrates the critical role that data architecture plays within the enterprise, including considerations for standards and applications, as well as technical and business drivers [14].

There is no one (i.e., single) data architecture paradigm that can fulfill all the needs of the fusion workflow within an enterprise. Furthermore, a fusion enterprise is an unlikely stand-alone implementation candidate, as Decision Fusion (DF) is generally not an objective in itself, but rather part of a larger mission within a System of Systems (SoS) Net Centric Operation (NCO). Therefore, the data architecture implementation complexities arise from the requirements for both horizontal and vertical integration within the enterprise. Horizontal integration includes interoperability with other business applications at the same operational layer as well as the need for a cohesive alignment (e.g. format, structure, units) of different Multi-Int collections to enable data enrichment (i.e., supplementing entity characterization from one source with attributes from other sources). Vertical integration requires the effective transition of application, interfaces, and transformation with other components upstream and downstream from the business processes and fusion workflow. Intertwined throughout the end-to-end fusion workflow, is the need to manage enterprise considerations such as metadata, access control, and backup and recovery strategies. Note that there is an important distinction in that what is being discussed is a layered Data Architecture - not a layered application architecture such as DIICOE. The critical distinction being that whereas a layered application architecture also provides portability and interoperability via standards and decoupling, its tiers represent a fairly homogenous set of functions; a classic example is the web server, application server, middleware, and database tiered architecture. Hence the issue: the database is usually at the backend, and is monolithic. What is being advocated here is the need for a tiered data architecture that is optimized for the layered vertical and horizontal functions of the fusion workflow, as the relationship between fusion algorithm and the structure of the underlying data it processes is tightly coupled – a one serve-all global database model won’t do. An enterprise presents additional challenges in that the capabilities of fusion technologies of the collection process, i.e., sensor platforms, and the COI mission objectives have evolved at different rates and in a fragmented fashion. Consider the evolution from the mission to detect, characterize, classify, and identify Ground Moving Target Indicators (GMTI); enrichment processes differ contingent on the type of mission, e.g., tracking a vehicle as a target vs. tracking an individual in a vehicle requires different cross-cueing considerations. This represents an impedance mismatch between Top-Down business drivers and Bottoms-Up sensor capabilities. Additionally, an enterprise is rarely statically designed and implemented in its entirety in one spiral. Rather, it is dynamically built upon the gradual integration of often disparate COI technologies with their own backend databases, which impute a significant data alignment and transformation complexity. In these enterprise cases, a SoS design approach should be favored over the more traditional Systems Engineering methodologies [15-17].

Fig. 3 depicts this layered data architecture. Upstream, data is collected from different sources and loaded into repositories specialized for heterogeneous data types such as Spatial, Multi-Media, Structured, e.g., relational, and Unstructured, e.g., free text, documents. This tier includes functions such as data cleansing and data integrity verification, as well as metadata tagging for enterprise discovery. The second tier serves as a pre-processing staging area, and provides services for transformation among different structures and formats, and mensuration alignment. The top tier includes the database integrated forensic and predictive processing for HL Fusion.
III. THE MULTI-INT REPOSITORY

This layer serves as the staging area for storing and managing operational data from Multi-Int sources (raw and post-processed data collections from platforms performing Level 0/1 Fusion). Technically, Multi-Int data are collections of different types or physical phenomena, e.g., multi-spectral imagery, radar-based GMTI, and their underlying data storage types include structured, unstructured data, spatial data, multimedia, and binary objects. In the broadest sense, all data is binary; notwithstanding, this refers to what are often proprietary data formats. Structured data can include relational, object relational and semantic technologies such as XML based Resource Description Framework (RDF). The advantages of DBMS technologies include the ability to effectively perform many key data management tasks such as integrity verification, metadata discovery, access control, replication, backup and recovery. Additionally, data stored in databases can also be optimized for processing performance using various indexing and partitioning techniques. For HL Fusion this presents an opportunity for applications to focus on analytics, and not be encumbered with the general data management functions. At the enterprise level, this also represents effective use of personnel, as data management can now be performed by Database Administrators, freeing the Fusion Analysts and Application Developers to focus on data exploitation.

A. Relational and Object-Relational Schemas

Relational data is arguably the most widely used structure for organizing entities with attributes in a table (row and columns), which allows for fusion via attribute association and manipulation of the data in a systematic way via a standard mechanism e.g. Structured Query Language (SQL). This is the traditional structure for On-Line Transaction Databases (OLTP). Variants of the structures such as data cubes are used in Analytical On-Line Processing (OLAP). As business entity modeling requirements became increasingly complex, databases structures began to include object-oriented paradigms, which allows entities to be defined by composition and inheritance.

B. XML and Semantic Schemas

XML evolved mainly from the need to provide interoperable data among Web Applications, and a need for easily extending structure definition. However, for what it makes up in interoperability and extensibility it lacks in performance and manipulation capability. It is not trivial to navigate an XML structure, and while there is ability to have DBMS embedded binary XML repositories, performance still lags its traditional relational cousin.

Whereas relational schemas associate entities through common attributes, semantic technologies extend this concept by providing specific ontologies for relationships whereby more complex associations can be automatically inferred. For example, defining relationship types such as Parent, Child, Sibling, can be used infer other relationship types such as Uncle/Aunt and Cousin; e.g. if \{A\} is Parent of \{B\}, and \{C\} is Sibling of \{A\} then \{C\} is “Uncle/Aunt” of \{B\}. The W3C includes SQL, XML, Resource Description Framework (RDF), and Web Ontology Language (OWL) standards, which enable interoperability throughout the enterprise. A key advantage of maintaining ontologies is that concepts among COIs can be disambiguated; e.g., a “Tank” within the context of a plane or car is a place to store fuel, whereas in the context of the Army, it can also mean an armored vehicle. Conversely, equivalent concepts with different names among COIs can be mapped via ontologies to preserve and represent the shared connotation. A major trade-off of XML based structures is that they do not easily lend themselves for performing other types of analytics, such as clustering, classification, or link analysis (e.g., Social Networking Analysis).

C. Unstructured Data

Unstructured data usually refers to free text – i.e., text that has not been parsed into individual entities and characterized; this can include text files of various formats e.g. MS Word, MS PowerPoint. By running classifier and clustering algorithms on unstructured data such as text documents, they can be categorized according to topic, or mined for entity extraction – this is particularly useful in mining for Open Source Intelligence (OSINT), e.g., Web articles and documents. Various databases integrate Support Vector Machines into their data mining tool set, which provides for powerful nonlinear classifiers, which can be tuned as more data becomes available (i.e., machine learning).

D. Spatial Data - Geographic Information Systems (GIS)

Entities with geospatial attributes are related in terms of spatial connectivity, contiguity, proximity and so forth. This is the domain of GIS, which provides a rich set of functionality that operates on spatial primitives such as points, lines and polygons, as well as more complex topologies 3-D geometries e.g., Triangular Irregular Network. GIS includes spatially enabled databases, which stores coordinate geometry of various topologies. Fig. 4 depicts geospatial layers, showing vector and raster type topologies. Images can be treated as raster layers, whereby cell values represent pixels, which can be overlayed with other cell-based data e.g., elevation. Additionally, geospatial layers can contain temporal information, which can be used to perform time-series trend analysis.

Figure 4. GIS Layered Topology
E. Binary Large Objects (BLOB)

The term binary object is used in a broad sense, as all data digitally stored is de facto binary. Here, it is meant as a generic catch-all representation for proprietary unstructured data that cannot be readily parsed, such as the binary representation of many Multi-Int Sensors, e.g., Signal Intelligence (SIGINT), Communications Intelligence (COMINT), Synthetic Aperture Radar data, whose formats are often proprietary and not be managed through the usual Multi-Media applications and interfaces. The advantages of BLOBs type is that they are capable of storing and managing files up to 128 Tera bytes - if stored within the DBMS itself.

F. Multi-Media

Databases are an ideal mechanism to manage Multi-Media such as video, audio, and imagery. While it is possible to also include geo-referenced video and imagery, this type of data are best managed through a GIS, as multi-media does not perform geospatial operations on its data. Multi-Media content usually consumes large amounts of disk space, span diverse industry standards, and are relatively unstructured, e.g., an image or video frame contains only pixel information. Metadata is usually stored in headers and other file areas according to specific standards. Applications using Multi-Media can leverage DBMS provided APIs which facilitate tasks such as loading, metadata extraction, and indexing for efficient content searching and retrieval.

Additionally, multimedia content managed by a DBMS provide the benefits of enterprise security, back-up and recovery, and the ability to locate content across the enterprise using relational, object, or SQL Multi Media (ISO standard) interfaces. The advantage is that this capability enables a fusion application to focus on algorithms such as Feature Extraction, Automated Target Recognition, or streaming using Real Time Streaming Protocol (RTSP), and delegate the media management to the DBMS.

IV. Data Transformation & Management

Data transformation services are critical for preparing Multi-Int data for analytic processes throughout the fusion workflow. Additionally, quality issues are an utmost importance, and should ideally be managed as close to the collection point as possible, as precision and accuracy cannot be enhanced by downstream activities. Other inconsistencies should be resolved as a pre-processing activity, such as Data Profiling, whereby inconsistencies such as spelling, duplicates, and blanks are systematically addressed. Other facets data management include transformation, whereby data is converted into the necessary input specifications (standards, structures, and formats) required by the fusion algorithms. The following subsections describe a selected set of activities that need to be managed efficiently and effectively. These activities may require specialized data technology, such as ETL, Data Cubes, and database gateways that enable integration of disparate COI content into the enterprise data warehouse repositories.

A. Data Ingest

Ingest should be done within a Demilitarized Zone (DMZ) reserved for the sole purpose of transferring data to and from external sources, including distributed nodes with heterogeneous (multi-vendor) databases – gateways provide the means to integrate multi-vendor data stores. The DMZ approach safeguards the operational data from becoming corrupt. All ingest tasks should be tagged with metadata that captures the pedigree and includes task time-stamps. Additionally, the DMZ serves as a security buffer zone to prevent unintended and unauthorized data propagation. Fine-grained access control and pull (publish) alerts should be implemented from this stage on to all other secondary staging areas, such as data warehouses and data marts, according to specific and assigned user roles and responsibilities.

B. Data Cleansing and Profiling

Data cleansing ensures the optimization of a data set for a particular algorithm. For example, how to treat missing values must be done in the context of understanding how the algorithm computes and is biased towards an attribute in the data set. Techniques for replacing values range from substituting with the mode, average, or other techniques, and it could change according the subset under analysis – i.e., one technique could be used for national level data, whereas another technique should be use for regional or local data (from a geographic granularity context). One of the most common issues is that of inconsistencies such as those that occur in spelling variants of the same entity or various coding attributes for the same meaning. For geospatial data sets, topology data cleansing becomes even more critical. Data sets that comprise polygon and line topology are especially prone to topological inconsistencies; an unclosed polygon will be interpreted as a line (arc) and not be flagged as an issue; this will produce in faulty results in fusion analytics such as buffer and overlay. One way of minimizing data integrity issues, is to design the data model and structure to be as conspicuous as possible, e.g., one layer would represent only polygons, arcs in this set would be obviously not belong there.
Profiling captures key characteristics of a data set, such as minimum and maximum values, modes, duplicates, blanks, and outliers. These characteristics are also useful in determining how a particular algorithm will perform against a data set; e.g., a sparse data set may execute faster in a market basket (affinity) analysis, whereas more dense sets will require additional processing time. Profiling tools can automatically correct some quality issues according to pre-defined rules (e.g., substituting blanks with a mode or average).

C. Data Alignment

Fusion by association (e.g., relating two separate entities stored in two separate database relational tables) and function (e.g., overlaying two geospatial layers) is based on being able to match common attributes between two or more entities, e.g., name, date, geo-referenced location, speed, height. Data alignment is the process whereby this is accomplished and comprises three key characteristics which are: (a) semantic equivalence, (b) measurable unit equivalence, and (c) format equivalence. Some semantic equivalence is usually encompassed in the data cleansing, where anomalies such as spelling errors or variations are resolved. Additionally, ontological equivalence is achieved through more advanced methods such as mapping disparate vocabularies among COIs using technologies such as Web Ontology Language (OWL).

Data alignment in geospatial layers is more complex, as transformation between map layers of different zones is necessary before they may be used for analysis. Results of geospatial algorithms may differ depending on the map projection used; e.g., a circle buffer may result in an oval and miss features on a map that is non-conformal (does not preserve shape). Accordingly, geospatial data must be aligned to the same projection and projection parameters as well as units. Hence, a well thought out Database Architecture can ease the burden of the Fusion analyst by enabling key preprocessing functions in the workflow staging areas.

D. Data Standards and Formats

A data standard is the specification for the logical representation of particular data entity, it may or may not specify a format as well, i.e., the underlying structure of the data, e.g., matrix, relational, XML. Hence, adherence to a particular standard may not necessarily imply interoperability. For example, STANAG 4607 for defines GMTI data format for radar observations, however it does so only at the presentation layer protocol. Additionally, contingent on the function data can be represented in several formats; appendix G of the GMTI implementation guide proposes an XML schema that maintains a one-to-one correspondence with its binary counterpart; the use of XML is more appropriate for integrating with enterprise applications using Web Services. This example highlights the fact that when using standards, interoperability is at best a matter of degree; and accordingly may require numerous steps of data transformation, best allocated to database components.

Format comprises both (a) data representation (e.g., Monday, July 9, 2012 vs. 7/9/12) and (b) its underlying structure implementation (e.g., XML vs. binary). Many fusion algorithms are developed within an application community, but independently of enterprise considerations, which means that they dictate the required format for processing. Social Network Analysis (SNA) is usually approached as a node-link (network) problem, and algorithms may require a matrix format to efficiently process the data; from a collection standpoint, SNA data may be stored using XML, RDF, or even relational formats, and thus requires transformation.

E. Metadata

Metadata is arguably the most critical element in Process Refinement (JDL Level 4). The traditional view is that metadata describes only the datasets, and comprises comprehensive information for data discovery and pedigree; especially useful for end-users across distributed nodes. However, the role of metadata extends beyond this scope to include algorithms and process metadata, including input, output, and process parameters and their descriptions. For example, many Support Vector Machines (SVM) algorithms allow for process parameters specification such as controlling whether a Linear or Gaussian kernel is used, the amount of memory allocated to optimize processing performance, and specifying the outlier rate to detect anomalies. One of the newer uses of including enterprise process metadata, is in the design of a fusion marketplace, whereby analysts can compare results among the community for a similar domain problems, and develop and fine-tune application to data heuristics.

F. Access Control

Access Control is part of the Confidentiality-Integrity-Availability (CIA) security triad, and with Access control, there is Identification, Authentication, and Authorization. The scope is very broad, but a critical component of security Enterprise Fusion, especially since it has to be managed and coordinate with at multiple tiers, such as fusion applications, and business policies. Accordingly, the role of database architecture pertaining security is extremely critical, as implementation for data security need to propagate down to the lowest level of data granularity and managed by the DBMS. It does serve the enterprise, if access control is only managed at the application layer, and the underlying data in the DBMS is unsecured. A DBMS provides means for locking tables, establishing role based access, and implementing fine-grain access control whereby individual rows and even cells can be secured by compartment tagging (e.g., or confidential, restricted, classified) or conditional tagging (e.g., only data belonging Agency A, or viewing data attributes between a certain range), also known as content dependent access control.

A DBMS provides functionality that enables implementation of specific security control models such as Bell-LaPadula (Confidentiality: no read-up, no write-down), or Biba (Integrity: no read-down, no write-up); other models include Clark-Wilson (separation of duties), and Brewer-and-Nash, which is especially designed to prevent conflicts-of-interest access.
V. EMBEDDED ANALYTICS

With the advent of database embedded analytics and specialized data type stores, such as On-Line Analytical Processing (OLAP) and RDF respectively, more fusion processing has permeated throughout architectural layers (e.g., presentation, application, middleware) down to the database layer. The traditional relational schema used for On-Line Transactional Processing (OLTP) is not well suited for all types of data analytics. This section presents an overview of database embedded analytic functions and their corresponding architecture components. Fig. 6 depicts selected data mining components, and presents a high-level view of how to thread them together – ETL and data backup and recovery among the fusion workflow stages is best managed by database technologies.

Data Marts (DM) are specialized subsets of an Enterprise Data Warehouse, and as such are well suited to manage domain specific data, which can be optimized for performance (e.g., refresh rates) and configured (e.g., security protocols) according to the need of a COI. DMs present a way of decoupling the data management of a particular mission from the enterprise operational stores.

Fig. 7 depicts the data cube, which enables the “slice and dice” analytics of DM content. Data cube dimensions represent views of interest (e.g., geographic, temporal, semantic), and their corresponding hierarchies represent levels of granularity (e.g., Geographic: Country, State, City; Temporal: Year, Quarter, Month). Hierarchies are by definition based on attributes which lend themselves to logical and numeric aggregation, e.g., total crimes by month and year, and by city and province – this allows the analyst to drill-down to tactical levels or roll-up to strategic levels of granularity.

What makes this architecture ideal, is that it enables organizing and performing analytics based on consistent levels granularity. A tactical COA cannot be developed at the lowest level of granularity, e.g., a city, from strategic data e.g., a country. For example, an anti-burglary COA for a specific city cannot be developed based solely on national level theft statistics. Time-series analysis can easily be incorporated in data-cubes. This analysis can be extended to compare trends among hierarchies, revealing underlying causal forces which may not be apparent by running classifiers and clusters at the highest level of granularity. For example the effect of accumulating a specific type of weapon upon a specific crime may be traced to a city by examining trends of weapon caches among the geospatial hierarchies. Most OLAP capabilities to not include inductive reasoning (i.e., computational learning), so it is not an end-to-end fusion solution, but rather plays a role in the discovery and profiling function of the HL Fusion workflow. However, many databases vendors offer powerful data mining analytics that can be integrated with data cubes.

A. Data Warehouses, Data Marts and the Data Cube

A Data warehouse (DW) consolidates content from operational databases and is optimized for multi-dimensional queries. The processes of loading data into a DW is especially well suited for Multi-Int fusion as specialized Extract-Transform-Load scripts can automate the process and be tailored for the multiple formats of the Fusion Level 0/1 data formats. The DW is a good example of form follows function, as data is usually denormalized into star and snowflake pattern schemas, which facilitate time-series, trend, and pattern analysis – as opposed to normalized data which is optimized for OLTP. The traditional DW has an underlying relational schema upon which dimensions are built to support the “Slice and Dice” analytical capability of OLAP. The implication is that this relational structure is not well suited for Multi-Int data such as geospatial layers. Notwithstanding, geospatial dimensions and hierarchies (e.g., Country, Province, City) can still be defined, and their non-coordinate geometry attributes can be stored within the data cube and mapped to their corresponding layers stored in specialized repositories such as ESRI’s Spatial Database Engine, which integrates seamlessly with DBMSs from multiple vendors, or Oracle Spatial.
B. Semantic Data Architectures

Semantic models describe vocabularies (ontologies) for a particular domain. This knowledge representation is based on the open W3C standard OWL which enables modeling formal syntax for semantics, i.e., rules defining meaning and context for domain specific entities and their relationships. A key value is that an ontology models can be extended to provide vocabulary equivalence mapping among COIs, i.e., two differently named terms which share the same meaning can be used for entity enrichment and association; conversely, equally named terms which have different context meanings can be disambiguated. OWL is based on RDF/XML; RDF is made of triples comprising a Subject (identifies the object being described), Predicate (the data element to which a value is assigned), and an Object (the actual value). A distinct advantage of this model is that the Subject is a Uniform Resource Identifier (URI), which in turn can be a Uniform Resource Locator (URL), thereby making it very suitable for processing Web Content.

Fig. 8 depicts a high-level semantic workflow, based on open standards such as OWL, RDF, XML, SQL, and SPARQL. The ontology mapping shows the overlap among COIs, which serves to enrich the context and processing capability via aggregation. OWL is mainly a processing technology; accordingly, tools for visualization such as Table Reports, Node-Link Graphs, Relevance Charts, and Timelines need to be used to facilitate interpretate of query and inference results.

C. Text Mining (Unstructured Data)

Relational DBMS were originally developed to process structured data, i.e., entities with explicit attributes, and with specific data types and formats e.g., numeric, character, date. Now databases include the ability to exploit unstructured data by extracting entities and their attributes from documents such as MS PowerPoint or Adobe PDF files, and categorizing these documents by themes. This process uses algorithms such as (a) rule-based, whereby an analyst manually defined rule-sets, or (b) supervised, by providing a training-set, and using Support Vector Machines (SVMs) to automatically generate rule-sets, a significant advantage when dealing with a large number of documents, and (c) unsupervised, using clustering algorithms which automatically create categories, bringing attention to themes not previously considered. By itself this capability is not revolutionary, but leveraging the embedded analytics provides a seamless integration to ingest and process documents from sources which are within the DBMS, the file system, or the Web. Additionally, embedded functions such as alerts, can be triggered when there is a match of interest on a theme, and subsequently combined with other text-mining functions such as feature extraction, latent semantic indexing or ontology mapping. DBMS embedded functions such as (a) Soundex, which allows phonetic matching (b) Fuzzy queries, which allows variances in spelling, and (c) proximity operators, which allow for context building, provide further flexibility to enhance the exploitation of unstructured data.

VI. ENTERPRISE INTEGRATION

So far the discussion has centered on the Data Architecture, however this is only one view of the enterprise. A Database Architecture should be designed and developed in lockstep, and reconciled with the other enterprise considerations such as its business, application, and technical models and views (reference Fig.2). In addition to a Fusion infrastructure, Data Architecture plays a key role in integrating with many traditional enterprise application architectures components, such as Service Oriented Architecture (SOA) infrastructure, Enterprise Service Bus (ESB), Cloud Computing, and Cross-Domain Solutions, whereby the CIA considerations of data exchanges among various security classification domains cannot be compromised. Integration with any of these components requires a trade-off analysis [16].

A distributed architecture often comprises many geographically dispersed COIs, each with their own fusion center. DBMS provide numerous support functions for this distributed paradigm. Integrated functions include the ability to execute federated queries, and setup conditional data push/pull (or pub/sub) rules, which can be based on complex operators such as geospatial criteria whereby an Area-of-Interest (AOI) serves as the execution trigger. It might be tempting to consider that an application layer architecture such as SOA/ESB services alone can fulfill all data management needs for Enterprise Fusion. While these paradigms are generally appropriate for metadata, messaging, or relatively
small amounts of XML data, applications servers are not the best place to perform bulk replication of Multi-Int data—they are not optimized for large data transfers, and require extensive business logic to address functions such as security and data integrity, and to reconcile incongruent transfers (e.g., which site is authoritative, does the first or the last entry take precedence). DBMS offers embedded replication functionality that addresses business rules reconciliation, and includes two-phase commits to ensure data integrity. Finally, scaling requires special configuration at the DBMS. Load balancing is more complex than at the Web, or Application Server Layers, as tuning data access/retrieval performance is best managed at the database tier, with technologies such as GRID and Real-Application Clusters, Partitioning, and special indexing.

VII. CONCLUSION

Although a database has been included in the JDL Fusion model since its inception (albeit, mainly as a data repository), the complexities that arise from the specialized structures required to support Multi-Int data, the trend for embedding analytics in a DBMS, and the need to integrate diverse business requirements from multiple COIs within the Enterprise, has given rise to Data Architecture (DA) as a key enabler for implementing a successful HL Fusion (HLF) System. The following are key highlights from this presentation:

- One data global repository/model/format is insufficient
- DBMS provides global services, e.g., security, backup
- DBMS offers a rich and integrated set of HLF analytics
- DA requires Horizontal and Vertical Integration

A layered architecture is an efficient approach for designing a decoupled architecture. What is not evident, is that the continuous transformation that data undergoes in the fusion workflow, necessitates horizontal integration as well: (a) the repository layer needs to be threaded to provide cohesive ingest and access of Multi-Int data, (b) the transformation layer needs to thread together the conversions among many diverse data structures, (c) and the HL application layer needs to thread together many different fusion models to build the overall mission-level COA. Additionally, the data design has to reconcile top-down (business) and bottoms-up (legacy implementation) drivers; issues that are traditionally not part of other types of layered architectures such as B2C (Business to Customer) on-line retailer.

So what is pragmatic about the approach presented herein? The novelty lies not in the development of a specific domain fusion algorithm (e.g., track fusion), whose application is usually tailored for a vertical solution—but rather, it is the formalization that Data Architecture plays a critical role in the design of a Enterprise Fusion System, and that this design can leverage an extensive toolset of commercially available components (i.e., reuse) with which to build it. It is pragmatic as it presents a roadmap for integrating the contributions of the Fusion Community (i.e., reuse) and for leveraging formal and robust disciplines such as Systems Engineering and Data Architecture (i.e., reuse) to deliver cohesive Net-Centric and Global Information Grid (GIG) Fusion solutions.

REFERENCES